

## Organizzazione modulare delle memorie

Vengono qui svolte alcune considerazioni relative alla struttura e all'organizzazione dei moduli di memoria, con particolare attenzione all'espansione dei moduli sia in termini di numero di locazioni sia in termini di numero di bit della parola di memoria. Si osservi che le temporizzazioni illustrate sono da considerare valide solo per *memorie statiche*; il caso delle memorie dinamiche è alquanto più complesso e non viene qui preso in considerazione.

### Elemento generico di memoria

Il generico elemento di memoria da un singolo bit (Fig. 1) possiede:

- Un ingresso DIN (*Data Input*) su cui viene applicato il dato da memorizzare.
- Un'uscita DOUT (*Data Output*) attraverso cui viene estratto il dato memorizzato; tale uscita è di tipo *tri-state*.
- Un ingresso WE (*Write Enable*) che abilita la scrittura del dato presente su DIN all'interno dell'elemento di memoria.
- Un ingresso RE (*Read Enable*) che abilita l'uscita DOUT e presenta su di essa una copia del dato memorizzato all'interno dell'elemento.

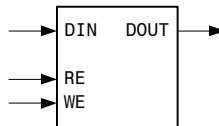


Fig. 1 – Elemento di memoria da 1 bit.

### Ciclo di lettura

La lettura di un dato (Fig. 2) avviene con la sequenza:

- All'istante  $t_0$  viene attivato Read Enable.
- All'istante  $t_1 > t_0$  DOUT esce dallo stato di alta impedenza e il dato memorizzato appare in uscita, e rimane stabile fin tanto che RE rimane attivo.
- All'istante  $t_2 > t_1$  RE viene riportato inattivo.
- All'istante  $t_3 > t_2$  l'uscita DOUT torna nello stato di alta impedenza.

Nel corso del ciclo di lettura WE deve restare inattivo, mentre sono irrilevanti i dati presentati su DIN.

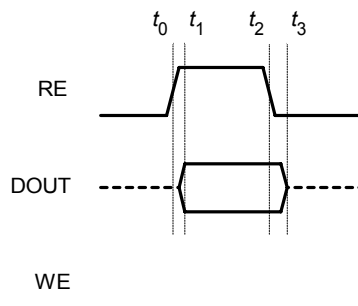


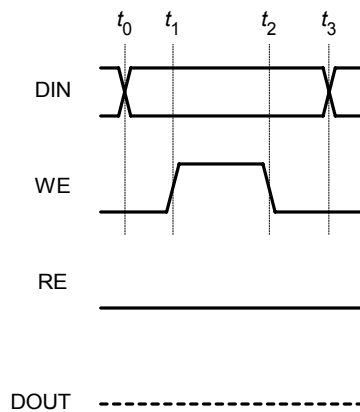
Fig. 2 – Ciclo di lettura.

### Ciclo di scrittura

La scrittura di un dato (Fig. 3) avviene con la sequenza:

- Il dato da scrivere viene presentato all'ingresso DIN (istante  $t_0$ ).
- All'istante  $t_1 > t_0$  viene attivato il Write Enable WE; l'intervallo  $t_1 - t_0$  è il *tempo di setup* ( $t_{\text{setup}}$ ) per la scrittura.
- All'istante  $t_2 > t_1$  WE viene riportato inattivo; l'intervallo  $t_2 - t_1$  è il *tempo di scrittura* propriamente detto ( $t_{\text{write}}$ ). A seconda della tecnologia con cui è realizzato l'elemento di memoria, il tempo di scrittura non deve scendere al di sotto di un dato valore affinché la scrittura avvenga correttamente.
- Fino all'istante  $t_3 > t_2$  il dato DIN deve restare stabile; l'intervallo  $t_3 - t_2$  è il *tempo di hold* ( $t_{\text{hold}}$ ).

È essenziale che DIN rimanga stabile entro tutto l'intervallo  $t_3 - t_0$ . Inoltre, durante il ciclo di scrittura, Read Enable rimane inattivo, e l'uscita DOUT rimane di conseguenza nello stato ad alta impedenza.

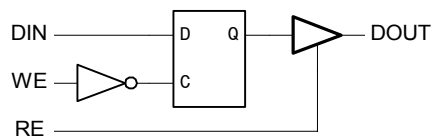


**Fig. 3** – Ciclo di scrittura.

### Realizzazione con flip-flop D

L'elemento di memoria da un bit può essere realizzato con un singolo flip-flop D (Fig. 4):

- Il dato di ingresso DIN viene applicato all'ingresso D del flip-flop.
- Il segnale WE viene applicato all'ingresso di clock del flip-flop mediante un invertitore, in modo che la scrittura avvenga in corrispondenza del fronte di discesa di WE (istante  $t_2$  in Fig. 3).
- Il dato in uscita DOUT viene prelevato dall'uscita Q del flip-flop attraverso un buffer tri-state, alla cui abilitazione è applicato RE.



**Fig. 4** – Flip-flop D come elemento di memoria da un bit.

### Realizzazione con flip-flop SR

L'elemento di memoria da un bit può essere realizzato anche con un singolo flip-flop SR (Fig. 5):

- Il dato di ingresso DIN viene applicato tramite una porta And all'ingresso S del flip-flop e anche, tramite un invertitore e una seconda porta And, all'ingresso R del flip-flop.
- Alle due porte And di cui sopra viene anche applicato il segnale WE.

- In tal modo le funzioni logiche che generano gli ingressi del flip-flop sono
 
$$S = DIN \cdot WE$$

$$R = \overline{DIN} \cdot WE$$
- Come per il flip-flop D, il dato in uscita DOUT viene prelevato dall'uscita Q del flip-flop attraverso un buffer tri-state, alla cui abilitazione è applicato RE.

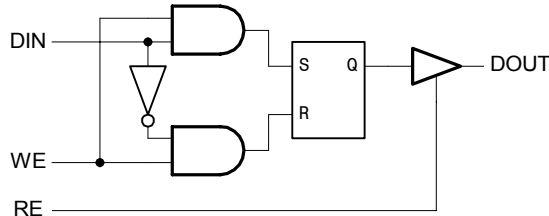


Fig. 5 – Flip-flop SR come elemento di memoria da un bit.

### Modulo di memoria con $N$ parole da 1 bit ( $NW \times 1$ )

Un modulo di memoria organizzato come  $N$  parole da  $M$  bit ciascuna (dove  $N$  è sempre una potenza di 2) viene identificato come  $NW \times M$ . Un modulo di memoria da 16 parole da 1 bit (identificato quindi come  $16W \times 1$ , Fig. 6) presenta:

- Una linea di ingresso DIN su cui viene applicato il dato da scrivere.
- Una linea di uscita tri-state DOUT dalla quale viene prelevato uno dei dati memorizzati.
- Una linea di ingresso WE per l'abilitazione alla scrittura del dato presente su DIN.
- Una linea di ingresso RE per l'abilitazione alla lettura dei dati memorizzati.
- 4 linee di ingresso,  $A_{0-3}$ , dette *indirizzo*, mediante le quali viene *selezionata* la parola entro cui scrivere il dato presente su DIN durante un ciclo di scrittura, ovvero dalla quale leggere informazioni durante un ciclo di lettura.

In generale, se il modulo contiene  $N$  parole, saranno necessarie  $\log_2 N$  linee di indirizzo.

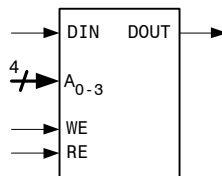
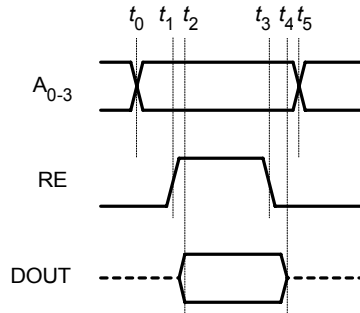


Fig. 6 – Modulo di memoria con 16 parole da 1 bit.

### Ciclo di lettura

Il ciclo di lettura è leggermente diverso da quello visto in Fig. 2, per via della presenza delle linee di indirizzo (Fig. 7):

- All'istante  $t_0$  viene applicato l'indirizzo sulle linee  $A_{0-3}$ , mediante il quale viene selezionata la parola che si desidera leggere.
- All'istante  $t_1$  viene attivato RE
- All'istante  $t_2$  la linea DOUT esce dallo stato ad alta impedenza e presenta il dato letto.
- All'istante  $t_3$  viene disattivato RE.
- All'istante  $t_4$  la linea DOUT torna allo stato di alta impedenza.
- All'istante  $t_5$  le informazioni sulle linee di indirizzo vengono modificate e si conclude così il ciclo di lettura.

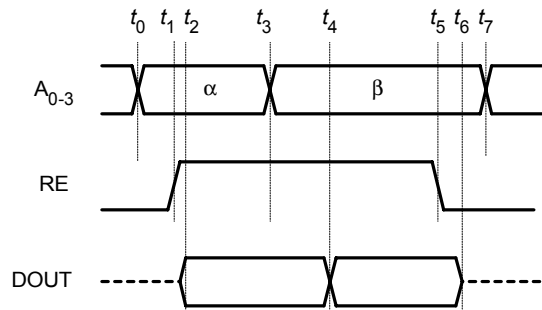


**Fig. 7** – Ciclo di lettura per il modulo 16W×1

Si noti che, dal momento che la lettura di un dato non è distruttiva, è ammissibile alterare l'indirizzo *anche se* RE è nel frattempo attivo (Fig. 8):

- All'istante  $t_0$  viene applicato l'indirizzo  $\alpha$  sulle linee  $A_{0-3}$ , mediante il quale viene selezionata la parola che si desidera leggere.
- All'istante  $t_1$  viene attivato RE
- All'istante  $t_2$  la linea DOUT esce dallo stato ad alta impedenza e presenta il dato contenuto nella locazione di indirizzo  $\alpha$ .
- All'istante  $t_3$  sulle linee  $A_{0-3}$  viene presentato l'indirizzo  $\beta$  (diverso da  $\alpha$ )
- All'istante  $t_4$  sulla linea DOUT si presenta il dato contenuto nella locazione di indirizzo  $\beta$ .
- All'istante  $t_5$  viene disattivato RE.
- All'istante  $t_6$  la linea DOUT torna allo stato di alta impedenza.
- All'istante  $t_7$  le informazioni sulle linee di indirizzo vengono modificate e si conclude così il ciclo di lettura.

L'intervallo di tempo  $t_4 - t_3$  è il *tempo di accesso*, ossia il tempo necessario per il completamento delle operazioni di selezione della parola all'interno del modulo.

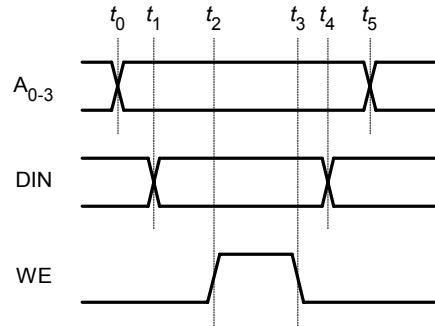


**Fig. 8** – Ciclo di lettura con modifica dell'indirizzo.

### **Ciclo di scrittura**

Il ciclo di scrittura nel modulo 16W×1 ha luogo con la sequenza:

- All'istante  $t_0$  viene applicato l'indirizzo sulle linee  $A_{0-3}$ , mediante il quale viene selezionata la parola entro cui si desidera scrivere.
- All'istante  $t_1$  viene applicato su DIN il dato che si desidera scrivere nella locazione il cui indirizzo appare sulle linee  $A_{0-3}$ .
- All'istante  $t_2$  viene attivata l'abilitazione alla scrittura WE.
- All'istante  $t_3$  viene disattivato WE.
- All'istante  $t_4$  il dato presente su DIN cessa di essere stabile.
- All'istante  $t_5$  l'indirizzo presente su  $A_{0-3}$  cessa di essere stabile



**Fig. 9** – Ciclo di scrittura per il modulo 16W×1.

Si noti come:

- Non è necessario che l'indirizzo venga reso stabile prima del dato; in altri termini, può anche essere  $t_0 > t_1$ ; ciò che deve essere garantito è che, prima dell'istante  $t_2$ , sia l'indirizzo sia il dato siano stabili per un intervallo almeno pari al tempo di setup:

$$t_2 - \max(t_0, t_1) \geq t_{\text{setup}}$$

- Analogamente, non è necessario che DIN cessi di essere stabile prima dell'indirizzo; in altri termini, può anche essere  $t_4 > t_5$ ; ciò che deve essere garantito è che, dopo l'istante  $t_3$ , sia l'indirizzo sia il dato rimangano stabili per un intervallo almeno pari al tempo di hold:

$$\min(t_4, t_5) - t_3 \geq t_{\text{hold}}$$

- Così come nel caso dell'elemento di memoria da 1 bit, il segnale WE deve restare attivo per un intervallo almeno pari al tempo di scrittura:

$$t_3 - t_2 \geq t_{\text{write}}$$

### **Realizzazione del modulo 16W×1**

Il modulo 16W×1 può essere realizzato come illustrato in (Fig. 10):

- 16 flip-flop D (di cui solo 3 riportati in figura) costituiscono le 16 parole da 1 bit del modulo.
- Un decodificatore con 16 uscite, controllato dalle linee di indirizzo  $A_{0-3}$ , provvede a distribuire il segnale WE ai flip-flop.
- Un multiplexer a 16 ingressi, anch'esso controllato dalle linee di indirizzo  $A_{0-3}$ , provvede a instradare le uscite Q dei flip-flop verso il buffer tri-state di uscita e tramite questo all'uscita DOUT.

Durante un ciclo di lettura:

- WE è inattivo e su tutti gli ingressi di clock dei flip-flop appare un livello logico fisso pari a 1, dunque non avvengono alterazioni di stato all'interno dei flip-flop stessi.
- L'indirizzo  $A_{0-3}$  applicato al multiplexer fa sì che l'uscita di un solo flip-flop (quello corrispondente all'indirizzo applicato) venga instradata verso il buffer tri-state.
- RE è attivo, per cui il dato presente all'uscita del multiplexer viene portato all'uscita DOUT.
- Quando RE diventa inattivo, DOUT viene forzato allo stato di alta impedenza.

Durante un ciclo di scrittura:

- Il dato di ingresso DIN è applicato in parallelo agli ingressi D di tutti i flip-flop.
- L'indirizzo  $A_{0-3}$ , applicato al decoder, fa sì che una ed una sola delle uscite (diciamo, l'uscita  $k$ -esima) di quest'ultimo sia attiva; di conseguenza, il segnale WE transita attraverso il  $k$ -esimo Nand e viene applicato al clock del  $k$ -esimo flip-flop. (L'inversione prodotta dal Nand ha lo scopo di fare in modo che la scrittura entro i flip-flop abbia luogo in corrispondenza del fronte di caduta di WE.)
- Di conseguenza il dato DIN viene scritto all'interno del  $k$ -esimo flip-flop.

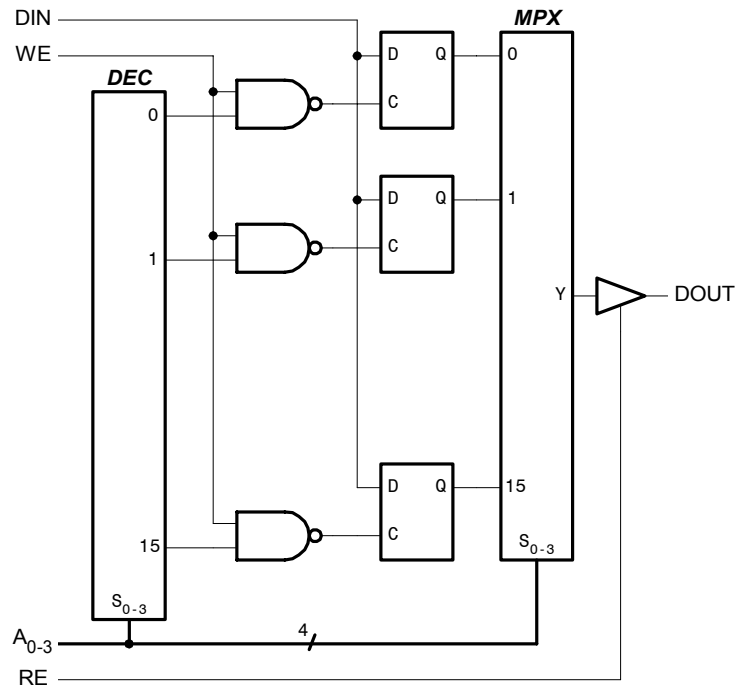


Fig. 10 – Realizzazione del modulo 16W×1.

### Osservazioni

L'implementazione del modulo 16W×1 appena descritta porta ad alcune considerazioni importanti ai fini della realizzazione di moduli di memoria più complessi:

- ❑ Le porte Nand all'ingresso dei flip-flop introducono un ritardo sul segnale di scrittura WE.
- ❑ All'aumentare del numero di parole del modulo, il multiplexer di uscita diventerà sempre più esteso.
- ❑ Anche il decoder di ingresso diventerà sempre più esteso all'aumentare del numero di parole del modulo, ma vedremo più avanti una tecnica che consentirà di ridurne le dimensioni.

Per ovviare a questi inconvenienti, è opportuno modificare la struttura dell'elemento di memoria da 1 bit in maniera che, oltre alle abilitazioni di lettura RE e di scrittura WE, vi sia in più un ulteriore segnale di abilitazione E comune (Fig. 11). In tal caso:

- ❑ Per eseguire una scrittura, WE ed E devono essere contemporaneamente attivi.
- ❑ Per eseguire una lettura, RE ed E devono essere contemporaneamente attivi.

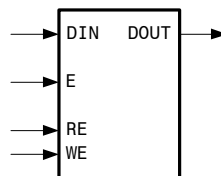


Fig. 11 – Nuova struttura dell'elemento di memoria.

### Realizzazione con flip-flop D

L'elemento di memoria così modificato può essere realizzato con un flip-flop D come illustrato in Fig. 12:

- ❑ Il buffer tri-state per DOUT è abilitato solo quando E ed RE sono contemporaneamente attivi.
- ❑ Il segnale WE è applicato al clock del flip-flop tramite un semplice invertitore.

- Quando  $E = 0$ , l'ingresso D del flip-flop vede la propria uscita Q: in tal modo, l'eventuale applicazione di WE non provoca una variazione di stato del flip-flop.
- Quando  $E = 1$ , l'ingresso D del flip-flop vede il dato applicato su DIN, così che un eventuale WE modifica di conseguenza lo stato del flip-flop.

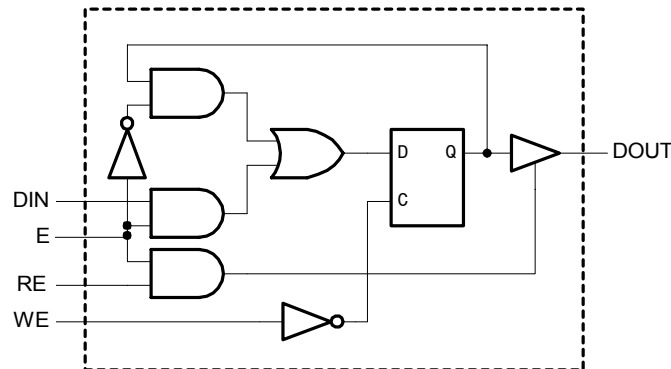


Fig. 12 – Realizzazione mediante flip-flop D.

### Realizzazione con flip-flop SR

La realizzazione con flip-flop SR è poi illustrata in Fig. 13:

- Il buffer tri-state per DOUT è abilitato solo quando E ed RE sono contemporaneamente attivi.
- Quando  $E = 0$  oppure quando  $WE = 0$ , gli ingressi al flip-flop sono  $S = R = 0$ , ossia il flip-flop non cambia di stato.
- Solo quando  $WE = E = 1$ , gli ingressi al flip-flop sono  $S = \text{DIN}$ ,  $R = \overline{\text{DIN}}$ .

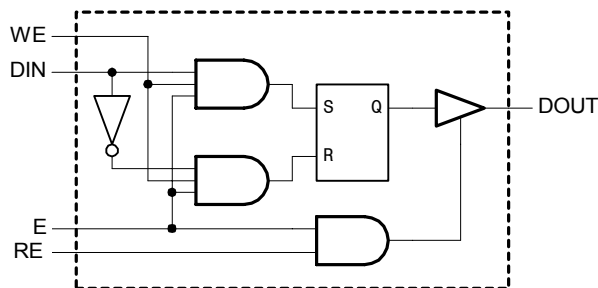


Fig. 13 – Realizzazione mediante flip-flop SR.

### Struttura del modulo $16W \times 1$

Utilizzando un tale elemento di memoria, il modulo  $16W \times 1$  può essere assemblato come illustrato in Fig. 14:

- Scompare il multiplexer di uscita, che viene adesso *distribuito* all'interno dei singoli elementi di memoria.
- Se il decoder usato possiede una sua propria abilitazione, questa può essere impiegata come abilitazione per l'intero modulo (tale caratteristica, come si vedrà più avanti, potrà essere sfruttata quando il modulo  $16W \times 1$  verrà utilizzato come elemento costituente di moduli di capacità maggiore).

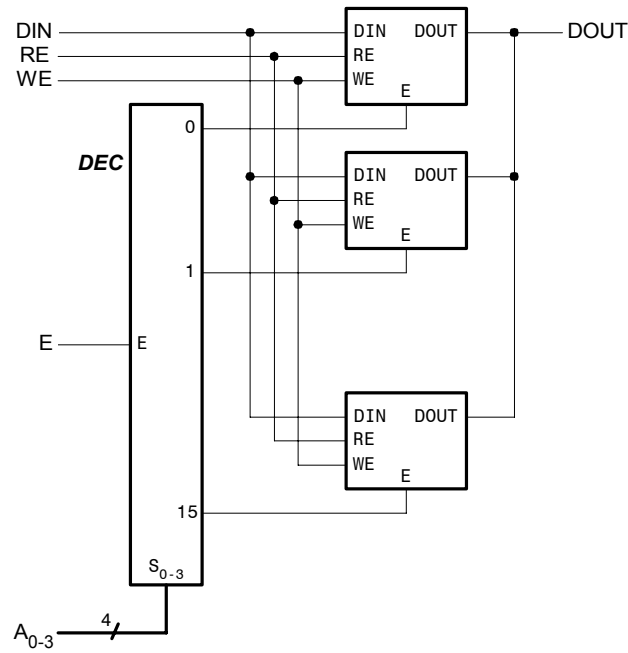


Fig. 14 – Modulo di memoria 16W×1.

### Organizzazione a matrice

Il modulo 16W×1 può essere organizzato in maniera da utilizzare decoder più piccoli. È sufficiente organizzare le celle di memoria in una matrice 4×4 (Fig. 15), ed impiegare 2 decoder per la selezione di una singola riga e, rispettivamente, di una singola colonna:

- Ad ogni incrocio riga-colonna, una porta And genera l'abilitazione per la cella.
- Le 4 linee di indirizzo vengono suddivise in due porzioni da 2 linee ciascuna: la prima porzione viene utilizzata per il controllo del decoder di riga, la seconda per il controllo del decoder di colonna. (È facile verificare che in tal modo, qualunque sia la configurazione binaria presente su  $A_{0-3}$ , ad ogni istante una ed una sola delle 16 celle risulta abilitata.)
- Le linee di ingresso DIN, RE, WE sono tutte connesse in parallelo tra di loro, così come le linee DOUT.
- Se i decoder sono dotati di ingressi di abilitazione, questi vengono connessi tra loro in parallelo e la linea di ingresso risultante costituisce l'abilitazione generale al modulo.

Con una tale organizzazione, in generale, un modulo  $NW \times 1$  viene organizzato come una matrice

- quadrata se  $N$  è pari
- rettangolare se  $N$  è dispari.

Nel primo caso:

- la matrice ha  $\sqrt{N}$  righe e  $\sqrt{N}$  colonne
- i decoder di riga e di colonna hanno in totale  $2\sqrt{N}$  uscite, anziché  $N$  come in Fig. 14.

Nel secondo caso, posto  $N = PQ$ :

- la matrice ha  $P$  righe e  $Q$  colonne
- i decoder di riga e di colonna hanno in totale  $P + Q$  uscite, anziché  $PQ$  come in Fig. 14.



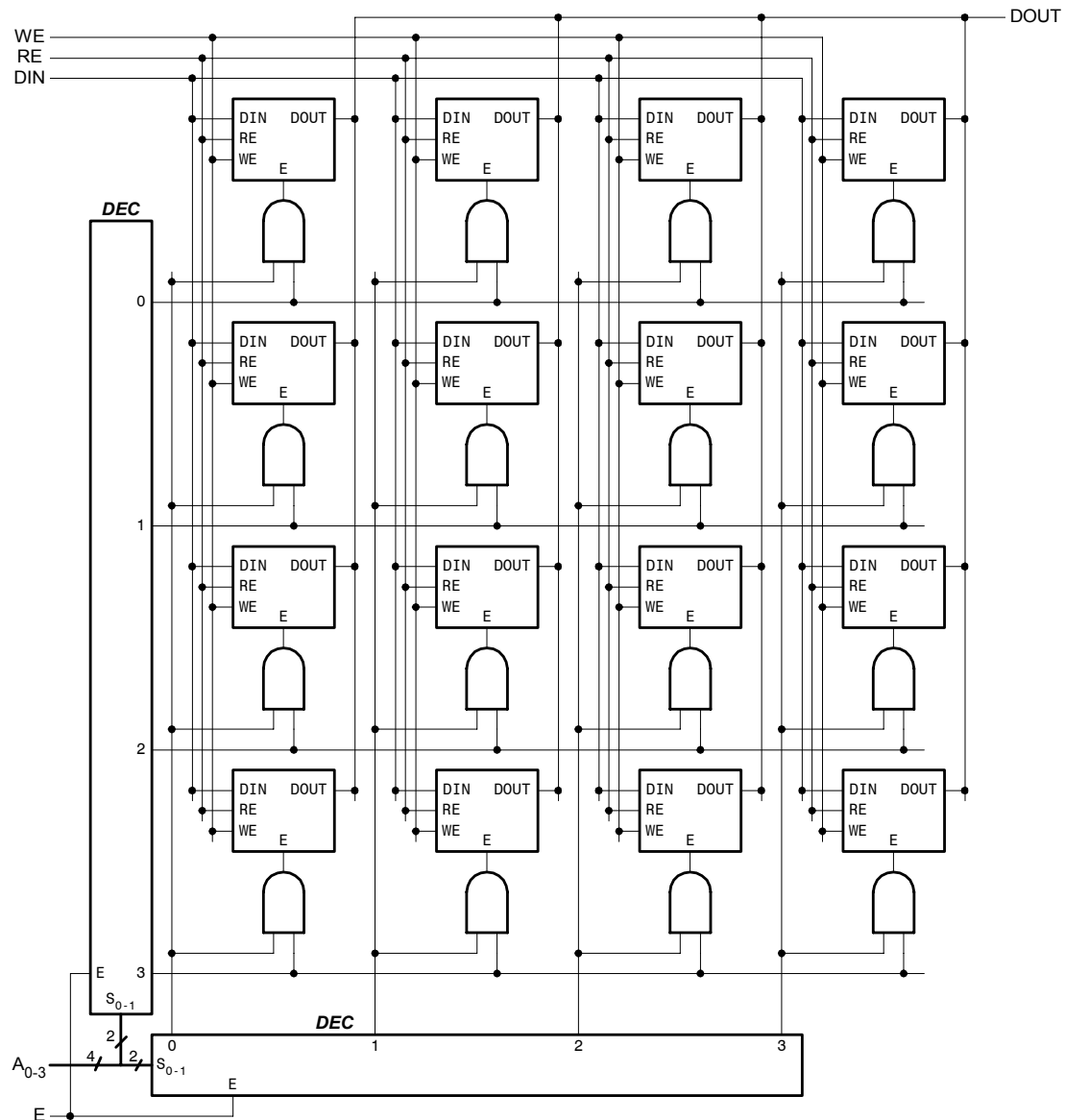


Fig. 15 – Modulo 16W×1 come matrice 4×4.

### ***Espansione del numero di locazioni***

Il modulo 16W×1 può a sua volta essere utilizzato come elemento per assemblare moduli di memoria di capacità maggiore. Ad esempio, un modulo 1KW×1 da 1024 parole da 1 bit può essere realizzato utilizzando 64 moduli 16W×1 disposti in matrice 8×8 (Fig. 16):

- Le linee DIN, RE, WE sono connesse in parallelo tra di loro, come pure le linee DOUT.
- Un decoder a 8 uscite seleziona la riga, e un decoder a 8 uscite seleziona la colonna.
- Le linee di indirizzo A0-9 vengono suddivise in tre porzioni: la prima da 3 linee controlla il decoder di riga, la seconda da 3 linee controlla il decoder di colonna, e la terza con le restanti 4 linee viene distribuita in parallelo a tutti i moduli 16W×1, essendo così utilizzata per la selezione della cella all'interno del modulo.
- Le abilitazioni dei decoder sono connesse in parallelo e formano l'abilitazione generale del modulo.

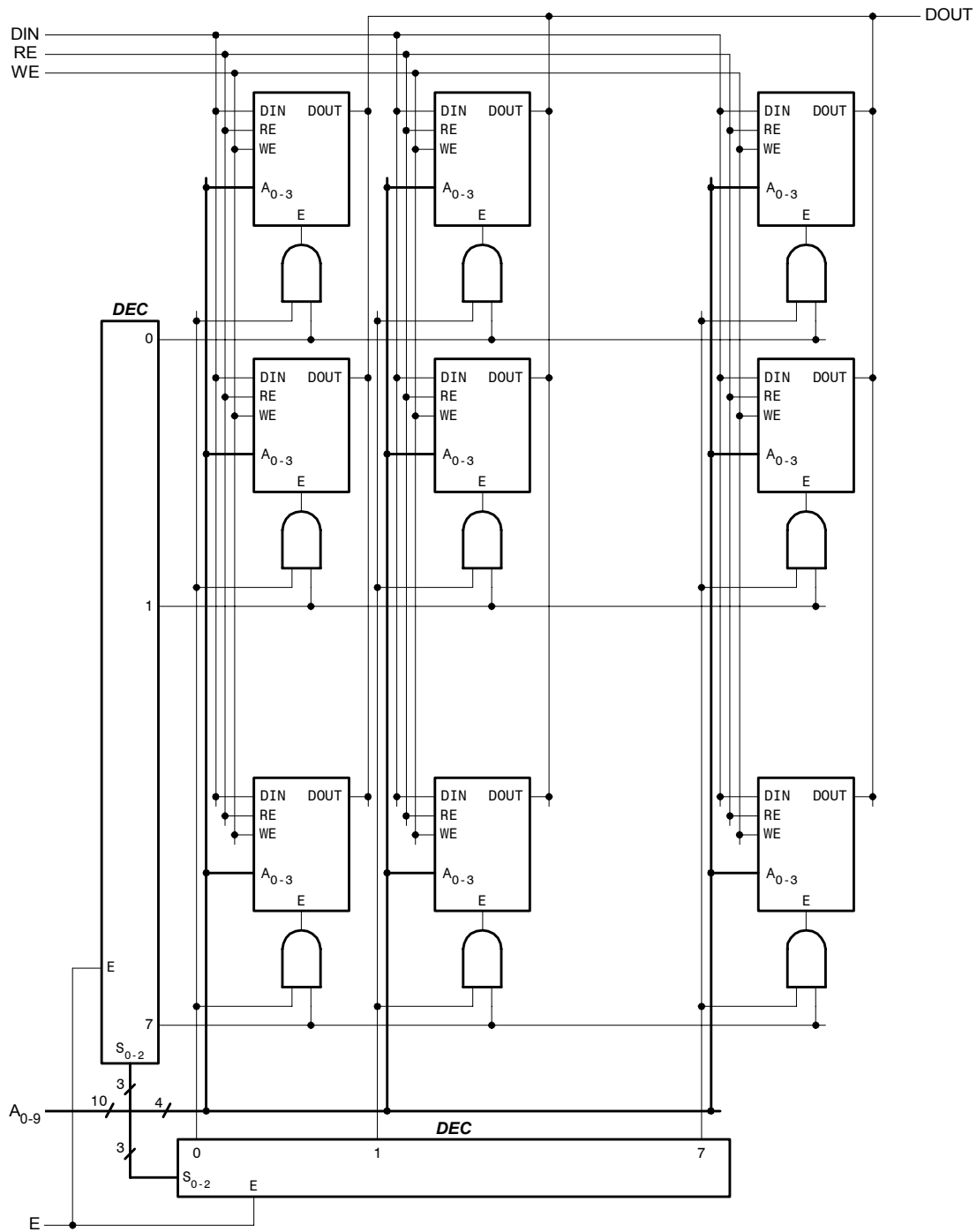


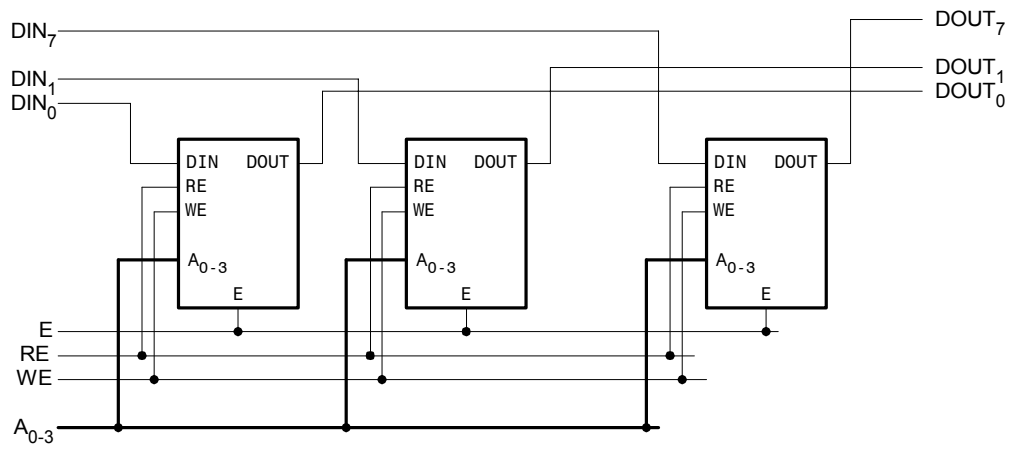
Fig. 16 – Modulo 16W×1.

### **Espansione della parola**

Per portare a  $M$  aumentare il numero di bit della parola di memoria, lasciando immutato il numero di locazioni, è sufficiente connettere  $M$  moduli da 1 bit. In () è illustrato l'uso di 8 moduli 16W×1 per la costruzione di un modulo 16W×8:

- Le linee RE, WE, E sono connesse in parallelo; in particolare, la linea E agisce da abilitazione generale del modulo.
- Ciascun bit delle linee DIN<sub>0-7</sub> di ingresso dati viene connesso all'ingresso DIN del rispettivo modulo 16W×1.

- ❑ Ciascun bit delle linee DOUT0-7 di uscita dati viene connesso all'uscita DOUT del rispettivo modulo 16W×1.
- ❑ Le linee A<sub>0-3</sub> di indirizzo sono connesse in parallelo a tutti i moduli.



**Fig. 17** – Modulo 16W×8.